



Dell Storage Center with Red Hat Enterprise Linux (RHEL) 6x Best Practices

Daniel Tan, Product Specialist
Dell Storage Applications Engineering

May 2015

Revisions

Date	Revision	Description	Author
Oct 2013	1.0	Initial release	Daniel Tan
Dec 2013	1.1	Refreshed for RHEL 6.4	Daniel Tan
May 2015	1.2	Introduce connectivity to Dell Storage SCv2x00	Daniel Tan

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2013-2015 Dell Inc. All Rights Reserved.

Dell, the Dell logo and the Dell badge are trademarks of Dell Inc.

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products.

Dell disclaims any proprietary interest in the marks and names of others.



Table of contents

Revisions.....	2
Executive summary	6
1 Overview.....	7
2 Managing volumes	8
2.1 Scanning for new volumes.....	8
2.1.1 Kernel versions 2.6 thru 2.6.9	8
2.1.2 Kernel versions 2.6.11 and newer	9
2.2 Partitions and filesystems	9
2.2.1 Partitions.....	9
2.2.2 Logical volume management	10
2.2.3 Volume labels and UUIDs for persistent device management	10
2.2.4 Creating a filesystem volume label	11
2.2.5 Existing filesystem volume label creation.....	11
2.2.6 Discover existing labels.....	11
2.2.7 /etc/fstab example	11
2.2.8 Swap space	12
2.2.9 Universally unique identifiers	12
2.2.10 About GRUB	13
2.2.11 ext4 and upgrading from an ext3 filesystem.....	13
2.2.12 ext4 and SCSI UNMAP (Free Space Recovery).....	14
2.3 Expanding volumes.....	15
2.3.1 Online expansion	15
2.3.2 Offline expansion	15
2.4 Volumes over 2TB.....	16
2.4.1 Creating a GPT partition	16
2.5 Removing volumes	17
2.5.1 Single volumes.....	17
2.5.2 Multipath volumes	17
2.6 Boot from SAN.....	18
2.7 Recovering from a boot from SAN Replay View volume.....	20
2.7.1 Red Hat Linux 6 and newer	20



3	Useful tools.....	21
3.1	The lsscsi command.....	21
3.2	The scsi_id command.....	21
3.2.1	Red Hat Linux 6 and newer.....	21
3.2.2	Red Hat Linux 5 and older.....	22
3.3	The /proc/scsi/scsi command.....	22
3.4	The dmesg command.....	23
4	Software iSCSI.....	24
4.1	Network configuration.....	24
4.2	Configuring RHEL.....	25
4.3	Configuring SLES.....	26
4.4	Scanning for new volumes.....	27
4.5	/etc/fstab configuration.....	27
4.6	iSCSI timeout values.....	27
4.7	Multipath timeout values.....	28
5	Server configuration.....	29
5.1	Managing modprobe.....	29
5.1.1	Red Hat Linux 6, SLES 11/10 and newer.....	29
5.1.2	Red Hat Linux 5, SLES 9 and older.....	29
5.1.3	Reloading modprobe and RAM disk (mkinitrd).....	29
5.1.4	Verifying parameters.....	30
5.2	SCSI device timeout configuration.....	30
5.3	Queue depth configuration.....	30
5.4	In single-path environments.....	31
5.4.1	PortDown timeout.....	31
5.5	In multipath environments.....	32
5.5.1	PortDown timeout.....	32
5.5.2	Multipath a volume.....	32
5.5.3	Multipath aliases.....	36
5.5.4	Storage Center device definition.....	37
5.6	Serial Attached SCSI.....	39
5.6.1	SAS drivers.....	39



5.6.2	SAS /etc/multipath.conf.....	39
5.6.3	FC/iSCSI & SAS.....	40
5.6.4	Identify SAS devices on Linux	40
5.6.5	Identify SAS devices on Dell SCv2000	41
5.6.6	Configured multipath	43
5.6.7	SAS queue depth	44
5.6.8	Boot from SAS	44
6	Performance tuning	45
6.1	Leveraging the use of multiple volumes.....	46
6.2	Understanding HBA queue depth	47
6.3	Linux SCSI device queue variables.....	48
6.3.1	Kernel IO scheduler	48
6.3.2	read_ahead_kb.....	48
6.3.3	nr_requests	48
6.3.4	rr_min_io.....	49
6.4	iSCSI considerations.....	49
7	The Dell Command Utility.....	51
7.1	Verifying Java, configuring and testing CompCU functions.....	51
7.2	Using CompCU to automate common tasks	53
7.2.1	Creating a single volume with CompCU	53
7.2.2	Creating a Replay and a Replay View with CompCU	54
7.2.3	Rapid deployment of multiple volumes with CompCU.....	54
A	Additional resources.....	55
B	Configuration details.....	56



Executive summary

Red Hat Linux® is an extremely robust and scalable enterprise-class operating system. Correctly configured using the best practices presented in this paper, the Red Hat Linux OS provides an optimized experience for use with the Dell™ Storage Center. These best practices include guidelines for configuring volume discovery, multipath, file system and queue depth management.

The scope of this paper discusses versions 5.9 thru 6.4 of the Red Hat Enterprise Linux platform with the Dell Storage Center SCOS version 6.x.x. Because there are often various methods in which to accomplish the tasks discussed, this paper is intended as a starting point of reference for end users and system administrators.

This guide focuses almost exclusively on the CLI (Command Line Interface) because it is often the most universally applicable across UNIX and Linux distributions.



1 Overview

The Storage Center provides Linux-compatible and SCSI-3 compliant disk volumes that remove the complexity of allocating, administering, using and protecting mission critical data. A properly configured Storage Center removes the need for cumbersome physical disk configuration exercises and management along with complex RAID configuration mathematics. The Storage Center also provides RAID 10 speed and reliability at the storage array layer so that volumes do not need to be mirrored on the Linux OS layer.

The full range of Linux utilities such as mirroring, backup, multiple file system types, multipath, boot from SAN, and disaster recovery can be used with Dell Storage Center volumes.



2 Managing volumes

Understanding how volumes are managed in Linux requires a basic understanding of the `/sys` pseudo filesystem. The `/sys` filesystem is a structure of files that allow interaction with various elements of the kernel and modules. While the read-only files store current values, read/write files trigger events with the correct commands. Generally, the **cat** and **echo** commands are used with a redirect as standard input verses opening them with a traditional text editor.

To interact with the HBAs (FC, iSCSI and SAS), commands are issued against special files located in the `/sys/class/scsi_host/` folder. Each port on a multiport card represents a unique HBA, and each HBA has its own `hostX` folder containing files for issuing scans and reading HBA parameters. This folder layout, files and functionality can vary depending on HBA vendor or type (for example, QLogic Fibre Channel, Emulex Fibre Channel, software-iSCSI based HBAs or Dell 12Gbps SAS HBAs).

2.1 Scanning for new volumes

Beginning with kernel versions 2.6.x and newer, the driver modules required for the QLogic 24xx/25xx-series HBAs and the Emulex HBAs have been included in the base kernel code. This kernel version correlates to Red Hat Linux versions 4 and newer as well as SUSE Linux Enterprise Server (SLES) versions 10 and newer. The following instructions apply to the default HBA driver modules. If the vendor (QLogic, Emulex) proprietary driver has been used, consult the vendor specific documentation for instructions and details.

A major overhaul of the SCSI stack was implemented between kernel versions 2.6.9 and 2.6.11. As a result, the instructions to scan for new volumes are different for both before and after 2.6.11 kernel code versions.

Rescanning all HBAs when mapping and discovering a new volume will not cause a negative impact.

Linux hosts cannot discover LUN ID 0 on the fly. LUN ID 0 can only be discovered during a boot and is reserved for the OS volume in boot from SAN environments. All other volumes should be associated to LUN ID 1 or greater.

2.1.1 Kernel versions 2.6 thru 2.6.9

The following rescan commands apply to QLogic and Emulex Fibre Channel HBAs as well as software initiator iSCSI ports. This rescan of all the `hostX` devices (i.e. Fibre Channel HBA and software initiator iSCSI ports) discovers new volumes presented to the Linux host.

```
# for i in `ls /sys/class/scsi_host/`; do echo 1 >>
/sys/class/scsi_host/$i/issue_lip; echo "- - -" >> /sys/class/scsi_host/$i/scan;
done
```

Standard output is not generated from the above commands. Discovered volumes are logged accordingly in `/var/log/messages` and in the `dmesg` utility.



2.1.2 Kernel versions 2.6.11 and newer

The following commands apply to QLogic and Emulex Fibre Channel HBAs as well as software initiator iSCSI ports. They initiate a rescan of all hostX devices (i.e. Fibre Channel HBA and software initiator iSCSI ports) and discover new volumes presented to the Linux host.

```
# for i in `ls /sys/class/scsi_host/`; do echo "- - -" >>
/sys/class/scsi_host/$i/scan; done
```

Standard output is generated from the above commands. Discovered volumes are logged accordingly in `/var/log/messages` and in the `dmesg` utility.

2.2 Partitions and filesystems

As a block-level SAN, the Storage Center volumes inherit and work with partition and filesystem schemes supported by the OS. Variables to consider when determining which partition and filesystem schemes to use are listed in sections 2.2.1 through 2.2.12.

2.2.1 Partitions

Partition tables are not required for volumes other than the boot drive. It is recommended to use Storage Center provisioned volumes as whole drives. This leverages the native strengths of the Storage Center in wide-striping the volume across all disks in the tier from which the volume is provisioned. Additionally, the use of partition tables may cause challenges when expanding the volumes capacity.

The following is an example of the output received while creating an ext4 filesystem on the device **/dev/vdb** without a partition table.

```
# mkfs.ext4 /dev/vdb
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
6553600 inodes, 26214400 blocks
1310720 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
800 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Writing inode tables: done
```



```
Creating journal (32768 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

This filesystem will be automatically checked every 34 mounts or 180 days, whichever comes first. Use `tune2fs -c` or `-i` to override.

2.2.2 Logical volume management

When LVM is applied and used to manage volumes on the Linux OS, it installs metadata (such as LVM signatures) to the volumes as unique identifiers. At the time that this paper was published, very few OS tools allow system administrators to directly manage LVM metadata. Mounting the Replay View volumes on the same host as the source volume is not recommended since it would result in duplicate LVM signatures.

It is recommended that Storage Center volumes be used in whole disk mode. LVM may still be used if it can provide features or benefits that are not already provided by the SAN layer.

At this time, the Red Hat Enterprise Server 5.4 built-in OS tool is the only method available to manage multiple and duplicate LVM signatures on the same host.

2.2.3 Volume labels and UUIDs for persistent device management

The Linux OS is capable of discovering multiple volumes in the Storage Center. The new disks are given device designations such as **/dev/sda** and **/dev/sdb** depending on the Linux OS discovery method used by the HBA ports connecting the server to the SAN.

Among other uses, **/dev/sdX** is used to designate the volumes for mount commands including mount entries in **/etc/fstab**. In a static disk environment, **/dev/sdX** works well for entries in the **/etc/fstab** file. The dynamic nature of Fibre Channel or iSCSI connectivity inhibits the Linux OS from tracking these disk designations persistently across reboots.

There are multiple ways to ensure that the volumes are referenced by their persistent names. This guide will discuss using volume labels or UUIDs (universally unique identifiers) which should be used with single-path volumes. Volume labels are exceptionally useful when scripting Replay recovery. An example involves mounting a Replay View volume of a production Replay, to a backup server. In this case, the Replay View volume may be referenced by its volume label without needing to explicitly identify which **/dev/sdX** device it was associated with. The volume label is metadata stored within the volume and will be inherited by volumes cut from the Replay.

Do not use volume labels in a multipath environment; they will not work. Multipath device names are persistent by default and will not change. Refer to section 5.5.3 titled, "[Multipath aliases](#)" for details.



2.2.4 Creating a filesystem volume label

The *mke2fs* and *mkfs.reiserfs* commands with the *-L* and *-l*LabelName flags erase previous filesystem tables, destroy pointers to existing files and create a new filesystem and a new label on the disk.

The examples below demonstrate use cases of creating a new file system with the label FileShare.

The process below will format the volume and destroy all data on that volume.

```
# mke2fs -j -L FileShare /dev/sdc
# mkfs.ext4 -L FileShare /dev/sdc
# mkfs.reiserfs -l FileShare /dev/sdc
```

2.2.5 Existing filesystem volume label creation

To add or change the volume label without destroying data on the volume, use the following commands while the filesystem is in a mounted state to avoid impact to the filesystem or existing I/O.

```
# e2label /dev/sdb FileShare
```

It is also possible to set the filesystem label using the *-L* option of the *tune2fs* command.

```
# tune2fs -L FileShare /dev/sdb
```

2.2.6 Discover existing labels

To discover a volume label, use the following command.

```
# e2label /dev/sde
FileShare
```

2.2.7 /etc/fstab example

The *LABEL=* syntax can be used in a variety of places including mount commands and the **GRUB** boot configuration files. Volume labels can also be referenced as a path for applications that do not recognize the *LABEL=* syntax. For example, the volume designated by the label FileShare can be accessed at the path **/dev/disk/by-label/FileShare**. A sample abstract from the **/etc/fstab** file is shown below.

Label=root	/	ext3	defaults	1 1
Label=boot	/boot	ext3	defaults	1 2
Label=FileShare	/share	ext3	defaults	1 2



2.2.8 Swap space

Swap space can only be labeled at the time it is enablement. This should not pose a problem, as no static data is stored in swap. The example below demonstrates how to apply a label to a swap partition.

```
# swapoff /dev/sda1
# mkswap -L swapLabel /dev/sda1
# swapon LABEL=swapLabel
```

The new swap label can be used in **/etc/fstab** as a normal volume label.

2.2.9 Universally unique identifiers

An alternative to volume labels is UUIDs. Although lengthy, they are static and safe for use anywhere. A UUID is assigned at filesystem creation.

A UUID for a specific filesystem can be discovered using the *tune2fs -l* command.

```
# tune2fs -l /dev/sdc tune2fs 1.39 (29-May-2006) Filesystem volume name:
    dataVol
Last mounted on:      <not available>
Filesystem UUID:      5458d975-8f38-4702-9df2-46a64a638e07 [snip]
```

An alternate way to discover the UUID of a volume or partition is to perform a long listing on the **/dev/disk/by-uuid** directory.

```
# ls -l /dev/disk/by-uuid total 0
lrwxrwxrwx 1 root root 10 Sep 15 14:11 5458d975-8f38-4702-9df2-
46a64a638e07 -> ../../sdc
```

The output above lists the UUID as 5458d975-8f38-4702-9df2-46a64a638e07.

Disk UUIDs can be used in the **/etc/fstab** file (as shown in the following example) or wherever persistent device mapping is required.

```
/dev/VolGroup00/LogVol100 / ext3 defaults 1 1
LABEL=/boot /boot ext4 defaults 1 1
UUID=8284393c-18aa-46ff-9dc4-0357a5ef742d swap swap defaults 0 0
```

As with volume labels, if an application requires an absolute path, the links created in the **/dev/disk/by-uuid** directory should work in most situations.



2.2.10 About GRUB

In addition to **/etc/fstab**, the GRUB boot configuration file should also be reconfigured to reference volume labels or UUIDs accordingly. The example below uses a volume label for the root volume (a UUID can be used in the same manner). Volume labels or UUIDs can also be used in resume syntax as needed for hibernation procedures (suspending to disk).

```
title Linux 2.6 Kernel root (hd0,0)
kernel (hd0,0)/vmlinuz ro root=LABEL=RootVol rhgb quiet initrd
(hd0,0)/initrd.img
```

2.2.11 ext4 and upgrading from an ext3 filesystem

Volumes previously created with an ext3 filesystem can be converted and upgraded to ext4. In the example below, a multipath volume formatted as ext3 and mounted to the **/testvol2** path is converted to ext4. To verify the integrity of the filesystem, a file is copied to the filesystem and an **md5sum** is generated. This **md5sum** is then compared after the conversion is complete and an **fsck** is run.

As with any partition or filesystem operation, some risk of data loss is inherent. Dell recommends capturing a Replay volume and ensuring good backups exist prior to executing any of the following steps. This procedure is not recommended for any root (/) or boot (/boot) filesystems on the Linux OS.

Unmount the filesystem and perform an **fsck** as part of this conversion process.

Some data is copied to **/testvol2** and a **checksum** is captured.

```
# df -h .
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/testvol2 9.9G  151M   9.2G   2% /testvol2
# mount | grep testvol2
/dev/mapper/testvol2 on /testvol2 type ext3 (rw)
# cp -v /root/rhel-server-5.9-x86_64-dvd.iso /testvol2/ 32
`/root/rhel-server-5.9-x86_64-dvd.iso' -> `/testvol2/rhel-server-5.9-x86_64-
dvd.iso'
# md5sum rhel-server-5.9-x86_64-dvd.iso > rhel-server-5.9-x86_64-dvd.md5sum
```

The conversion to **ext4**, including running an **fsck**, proceeds as:

```
# tune4fs -O flex_bg,uninit_bg /dev/mapper/testvol2 tune4fs 1.41.12 (17-May-
2010)
Please run e4fsck on the filesystem.
# umount /testvol2
# e4fsck /dev/mapper/testvol2 e4fsck 1.41.12 (17-May-2010)
One or more block group descriptor checksums are invalid.      Fix<y>? yes
Group descriptor 0 checksum is invalid.      FIXED.
<SNIP>
Adding dirhash hint to filesystem.
```



```

/dev/mapper/testvol2 contains a file system with errors, check forced. Pass 1:
Checking inodes, blocks, and sizes
Pass 2: Checking directory structure Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/mapper/testvol2: ***** FILE SYSTEM WAS MODIFIED *****
/dev/mapper/testvol2: 13/1310720 files (0.0% non-contiguous), 1016829/2621440
blocks

```

Finally, the volume is remounted as an **ext4** filesystem and the **md5sum** is verified.

```

# mount -t ext4 /dev/mapper/testvol2 /testvol2
# cd /testvol2
# df -h .
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/testvol2 9.9G  3.8G  5.7G  40% /testvol2
# mount | grep testvol2
/dev/mapper/testvol2 on /testvol2 type ext4 (rw)
# md5sum rhel-server-5.9-x86_64-dvd.iso
1a3c5959e34612e91f4a1840b997b287 rhel-server-5.9-x86_64-dvd.iso
# cat rhel-server-5.9-x86_64-dvd.md5sum
1a3c5959e34612e91f4a1840b997b287 rhel-server-5.9-x86_64-dvd.iso

```

The **/etc/fstab** file should be updated accordingly to reflect the new **ext4** filesystem used to mount this volume in the future.

2.2.12 ext4 and SCSI UNMAP (Free Space Recovery)

The **ext4** file system in RHEL 6 supports the **SCSI UNMAP** commands, which allow for storage space to be reclaimed on a Storage Center (version 5.4 or newer) and maintenance of thinly provisioned volumes.

With RHEL 6, **SCSI UNMAP** traverses the entire storage and I/O stack through the Device Mapper multipath daemon and then to the Fibre Channel or iSCSI layer. To achieve this functionality, the *discard* flag must be issued when mounting the filesystem or placed in the **/etc/fstab** file accordingly.

```
# mount -o discard /dev/mapper/mpathb /mnt/mpathb
```

For more information, read the blog titled, "Native Free Space Recovery in Red Hat Linux" on Dell TechCenter at

<http://en.community.dell.com/techcenter/b/techcenter/archive/2011/06/29/native-free-space-recovery-in-red-hat-linux.aspx>

Additional documentation is available from the Red Hat Customer Portal at

<https://access.redhat.com/site/solutions/393643>



2.3 Expanding volumes

Attempting to expand a file system that is on a logical or primary partition is not recommended for Linux users. Expanding a file system requires advanced knowledge of the Linux system and should only be done after careful planning and consideration. This includes making sure that valid backups exist of the file system prior to performing volume expansion steps.

Expanding a file system that resides directly on a physical disk, however, can be done.

As with any partition or filesystem operation, there is some risk of data loss. Dell recommends capturing a Replay volume and ensuring good backups exist prior to executing any of the following steps.

2.3.1 Online expansion

Red Hat versions 5.3 and newer volumes can be expanded without unmounting the volume.

1. Expand the volume on the Storage Center.
2. Rescan the drive geometry (if multipath, rescan each path).

```
# echo 1 >> /sys/block/sdX/device/rescan
```

3. For multipath volumes, resize the multipath geometry.

```
# multipathd -k"resize map <devicename>"
```

Note:

- a. where multipath device is /dev/mapper/<devicename>
- b. there is no space between the -k parameter and the command string

4. Resize the filesystem.

```
# resize2fs -p /dev/path
```

2.3.2 Offline expansion

Prior to kernel version 2.6.18-128, volume geometry cannot be updated on the fly with volumes in a mounted state. To address volume expansion needs in this scenario, refer to the procedure outlined in this section.

This procedure only applies to volumes which have no partition table applied. This requires unmounting the volume but does not require a server reboot.

1. Expand the volume on the Storage Center.
2. Stop services and unmount the volume.
3. If multipath is running, flush the multipath definition.

```
# multipath -f volumeName
```

4. Rescan the drive geometry (for each path if multipath).



- ```
echo 1 >> /sys/block/sdX/device/rescan
```
5. For multipath volumes, recreate definition.  

```
multipath -v2
```

or

```
service multipathd reload
```
  6. Run **fsck**.  

```
fsck -f /dev/sdX
```
  7. Resize the file system.  

```
resize2fs -p /dev/sdX
```
  8. Mount the filesystem and resume services.

## 2.4 Volumes over 2TB

Linux will discover volumes larger than 1PB, but there are limitations to the partitions and filesystems that can be created on volumes of that capacity. The various Linux filesystems (such as **ext3**, **ext4**, **xfs**, **zfs** and **btfs**) have specifications which vary over time. Consult the appropriate Linux distribution documentation to determine the thresholds and limitations of each filesystem type. On x86-64bit machines, the largest supported **ext3** filesystem is just under 8TB. However, MBR partition tables (the most common and default for most Linux distributions) can only support partitions under 2TB.

The easiest way around this limitation is to use the whole volume/disk instead of applying a partition table to the volume. The entire volume/disk can be formatted with the filesystem of choice and mounted accordingly. This is accomplished by running **mkfs** on the device without applying a partition table.

The alternative to whole volume/disk usage is applying a GPT partition table instead of the traditional MBR system. GPT support is native in RHEL 6/5, SUSE Linux Enterprise Server (SLES) 11/10, and many other modern Linux distributions.

### 2.4.1 Creating a GPT partition

After the volume has been created and mapped, rescan for the new device. Then follow the example below to create a GPT partition on the device. In this case, the volume is 5TB in size and represented by `/dev/sdb`.

1. Invoke the **parted** command.  

```
parted /dev/sdb
```
2. Run the following two commands inside of the **parted** command, and replace *5000G* with the volume size needed.



```
> mklabel gpt
> mkpart primary 0 5000G
```

3. Finally, format and label the new partition.

```
mkfs.ext3 -L VolumeName /dev/sdb1
```

## 2.5 Removing volumes

The Linux OS stores information about each volume presented to it. Even if a volume is in an unmapped state on the Storage Center, the Linux OS will retain information about that volume until the next reboot. If the Linux OS is presented with a volume from the same target using the same LUN ID prior to any reboot, it will reuse the old data about that volume. This may result in complications, misinformation and mismanagement of the volumes, and potentially cause data loss in the environment.

It is therefore a recommended best practice to always unmount, remove and delete all volume information on the Linux OS after the volume is deemed no longer in use. This is non-destructive to any data stored on the actual volume itself, just the metadata about the volume stored by the OS (volume size, type, etc.).

### 2.5.1 Single volumes

First, unmount filesystems associated with this volume and determine the volume device file name (for example **/dev/sdc**). Issue the command shown below for each volume identified for removal. Finally, unmap the volume from the Storage Center.

```
echo 1 > /sys/block/sdc/device/delete
```

Edit the **/etc/fstab** file to remove any references to the volume and associated filesystem as required. This ensures that no attempt is made to mount this filesystem at the next boot.

### 2.5.2 Multipath volumes

The process for removing multipath volumes is similar to removing single volumes with the addition of a few steps.

With multipath volumes, each **/dev/mapper** device is represented by at least one device file as shown below in **bold**.

```
multipath -ll
LUN02_View (36000d31000006900000000000000014f2) dm-8 COMPELNT,Compellent Vol
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
 |- 6:0:4:3 sdm 8:192 active ready running
 `-- 7:0:6:3 sdn 8:208 active ready running
[snip]
```



After unmounting filesystems associated with the volume, edit the **/etc/multipath.conf** file to remove references to the volume WWID, aliases, and other identifiers. Issue the command below to force the multipath daemon to reload its configuration.

```
service multipathd reload
```

Remove the multipath backing device files by entering:

```
echo 1 > /sys/block/sdm/device/delete
echo 1 > /sys/block/sdn/device/delete
```

Finally, unmap the volumes from the Storage Center.

Edit the **/etc/fstab** file to remove any references to the volume and associated filesystem as required. This ensures that no attempt is made to mount this filesystem at the next boot.

## 2.6 Boot from SAN

Red Hat Linux versions 5.1 and newer have the ability to perform an OS installation directly to a multipath volume. This greatly simplifies the procedure to achieve multipath boot from SAN functionality.

To achieve this functionality, the identified boot from SAN volume needs to be presented to the Linux system as LUN ID 0 and mapped over 2 or more paths as discussed in the section 5.5.2 titled, "[Multipath a volume](#)". A LUN ID 0 mapping is accomplished during the process of mapping a volume to the Server Object. In the **Advanced Settings** page, select **Map volume using LUN 0** as shown in Figure 1, and then click **Continue**.



[Back](#)
[Return](#)
[Quit](#)
[Advisor](#)

### Select LUN

☒ Map volume using LUN 0 (this is usually reserved for boot volumes).  
☐ Use LUN  when mapping the selected volume to the selected server.  
☒ Use the next available LUN if the preferred LUN is unavailable.

### Restrict Mapping Paths

☐ Only map using specified server ports:

|                                     | Type | Server Port      | Status | Connected Controller |
|-------------------------------------|------|------------------|--------|----------------------|
| <input checked="" type="checkbox"/> | FC   | 21000024FF27D8CA | Up     | 5000D31000006505     |
| <input checked="" type="checkbox"/> | FC   | 21000024FF27D8CB | Up     | 5000D31000006505     |

☐ Map to controller  if possible

### Configure Multipathing

Maximum number of paths allowed:  OS Default: Unlimited

### Configure Volume Use

☐ The selected volume should be presented as read-only to the selected server.

Continue

Figure 1 Select **Map volume using LUN 0**

Red Hat Linux versions 5.9 and older require issuing the **mpath** parameters at the boot prompt during the boot time (for example, `boot: linux mpath`). With Red Hat Linux 6.x and newer, the **mpath** parameter is assumed and therefore not required.

Proceed through the installation and select the **mapper/mpath0** device as the installation target when prompted. Complete the installation and then reboot. Ensure that the **multipathd** service is started during boot time. To verify the status of **multipathd**, issue the command below.

```
service multipathd status
multipathd (pid 1830) is running...
```

To start the **multipathd** service, and ensure that it starts during boot time, issue the following commands. Also verify that the **/etc/multipath.conf** file exists and is configured accordingly.

```
service multipathd start
chkconfig --levels 345 multipathd on
```

## 2.7 Recovering from a boot from SAN Replay View volume

When Replay View volumes of a boot from SAN Storage Center are presented to the same host, they are unbootable. This is because the volume serial number of the Replay View volume is different from the original boot from SAN volume. The procedures outlined below may be used to circumvent some of these limitations with RHEL 6x hosts.

### 2.7.1 Red Hat Linux 6 and newer

1. Create a Replay View volume from the boot from SAN Replay.
2. Map the Replay View volume to another Linux host that is able to interpret and mount the base file system type.
3. Mount the **/boot** and **/** devices of this Replay View volume (**/dev/sdc** in this case).

```
mkdir /mnt/hank-boot
mkdir /mnt/hank-root
mount /dev/sdc1 /mnt/hank-boot
mount /dev/sdc3 /mnt/hank-root
```

4. Determine the WWID of the Replay View volume. Consult section 3.2 titled, "[scsi\\_id](#)" for details.
5. Update the **/etc/multipath.conf** file to use the new WWID value.

```
vi /mnt/hank-root/etc/multipath.conf
```

6. Update the **/etc/fstab** file. RHEL6 does not require this since it uses the UUID to mount.
7. Update the **/etc/grub.conf** file. RHEL6 does not require this since RHEL6 uses the UUID to locate the root volume. Also, the **initramfs/initrd** already contains the modules needed.
8. Map the volume to the original host as LUN ID 0 and boot the host.

**Note:** This causes the system to return an "invalid multipath tables" error.

9. Fix the multipath error.

```
vi /etc/multipath/bindings -> Comment out the old entries. Add the entry
for the new device.
vi /etc/multipath/wwids --> Add the new device to the file.
```

10. Create a new **initramfs** file.

```
dracut -v /boot/<filename>
```

11. Update **/etc/grub.conf** to use the new **initrd** file. Create a new entry and retain the old entry for a recovery option.
12. Reboot the system using the new **initrd** file. Verify that the multipath is working as normal.



## 3 Useful tools

Determining which Storage Center volume correlates to specific Linux device files can be tricky. Use the following built-in tools to simplify this process.

### 3.1 The `lsscsi` command

The **`lsscsi`** command is a tool that parses information from the **`/proc`** and **`/sys`** pseudo filesystems into a simple human readable output. Although it is not included in the base installs for either Red Hat 5 or SLES 10, it is in the base repository and can be easily installed.

```
lsscsi
[6:0:4:1] disk COMPELNT Compellent Vol 0505 /dev/sdf
[6:0:4:2] disk COMPELNT Compellent Vol 0505 /dev/sde
[6:0:4:3] disk COMPELNT Compellent Vol 0505 /dev/sdm
[6:0:5:1] disk COMPELNT Compellent Vol 0505 /dev/sdg
[snip]
```

As expected, the output shows two drives from the Storage Center and three front end ports that are visible but not presenting a LUN ID 0. There are multiple modifiers for **`lsscsi`** which provide more detailed information.

The first column above shows the **`[host:channel:target:lun]`** designation for the volume. The host number corresponds to the local HBA hostX device file that the volume is mapped to. The channel number is the SCSI bus address which is always zero (0). The target number correlates to the Storage Center front end ports (targets). Finally, the last number is the LUN ID where the volume is mapped.

### 3.2 The `scsi_id` command

The `scsi_id` command can be used to report the WWID of a volume and is available in all base installations. This WWID can be correlated to the serial number reported for each volume in the Storage Center GUI.

#### 3.2.1 Red Hat Linux 6 and newer

The following code snippet may be used as-is without warranty or may be adapted to the needs of the respective environment. This code applies the **`scsi_id`** command using its RHEL 6 syntax structure and displays a table correlating **`sdX`** device names to their respective WWID.

```
for i in `cat /proc/partitions | awk {'print $4'} | grep sd`
do
echo "Device: $i WWID: `scsi_id --page=0x83 --whitelisted --device=/dev/$i`"
done | sort -k4
```



### 3.2.2 Red Hat Linux 5 and older

The following code snippet may be used as-is without warranty or may be adapted to the needs of the respective environment. This code applies the **scsi\_id** command using its RHEL 5 syntax structure and displays a table correlating **sdX** device names to their respective WWID.

```
for i in `cat /proc/partitions | awk {'print $4'} | grep sd`
do
echo "Device: $i WWID: `scsi_id -g -u -s /block/$i`"
done | sort -k4
```

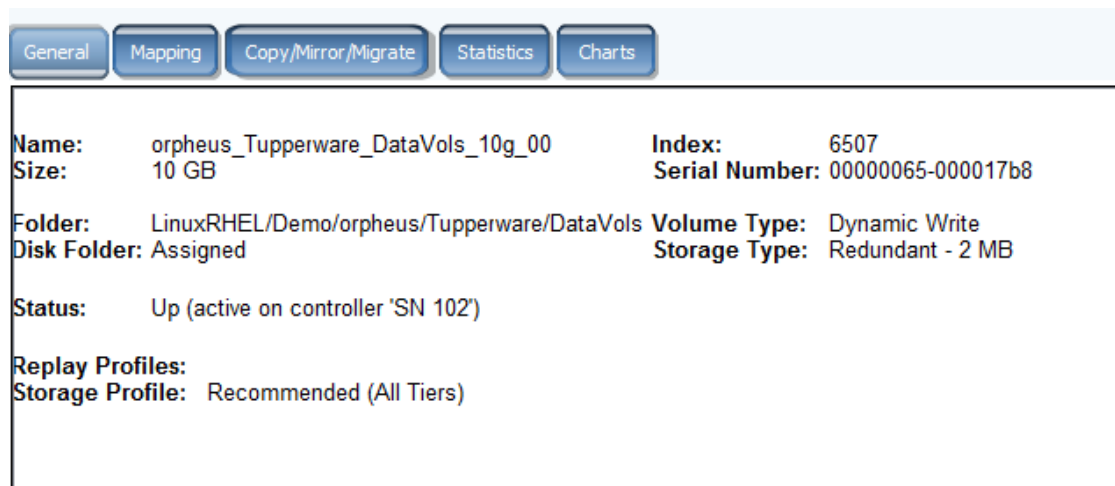


Figure 2 Observe the Serial Number of the volume: **Orpheus\_Tupperware\_DataVols\_10g\_00**

The first part of the WWID is the Storage Center unique vendor ID, the middle part is the controller number in HEX and the last part is the unique serial number of the volume. To ensure proper correlation in environments with multiple Storage Center arrays, be sure to verify the respective controller numbers as well.

**Note:** The only scenario where two volume serial numbers would not correlate is if a Copy Migrate function has been performed. In this case, a new serial number is assigned to the new volume on the Storage Center. The old WWID/serial number is used to present this new volume to the server in order to avoid any connectivity or I/O disruption.

## 3.3 The /proc/scsi/scsi command

The **/proc/scsi/scsi** file provides information about volumes and targets on Linux systems that do not have the **lsscsi** toolset installed.

```
cat /proc/scsi/scsi
Attached devices:
Host: scsi8 Channel: 00 Id: 00 Lun: 00
Vendor: PLDS Model: DVD-ROM DS-8D3SH Rev: HD51
Type: CD-ROM ANSI SCSI revision: 05
```



```
Host: scsi7 Channel: 00 Id: 05 Lun: 00
 Vendor: COMPELNT Model: Compellent Vol Rev: 0505
 Type: Direct-Access ANSI SCSI revision: 05
Host: scsi6 Channel: 00 Id: 05 Lun: 200
 Vendor: COMPELNT Model: Compellent Vol Rev: 0505
 Type: Direct-Access ANSI SCSI revision: 05
Host: scsi6 Channel: 00 Id: 07 Lun: 00
 Vendor: COMPELNT Model: Compellent Vol Rev: 0505
 Type: Direct-Access ANSI SCSI revision: 05
[snip]
```

## 3.4 The dmesg command

The output from **dmesg** is useful for discovering which device name is assigned to recently discovered volumes.

```
SCSI device sdf: 587202560 512-byte hdwr sectors (300648 MB)
sdf: Write Protect is off sdf: Mode Sense: 87 00 00 00
SCSI device sdf: drive cache: write through
SCSI device sdf: 587202560 512-byte hdwr sectors (300648 MB)
sdf: Write Protect is off sdf: Mode Sense: 87 00 00 00
SCSI device sdf: drive cache: write through sdf: unknown partition table
sd 0:0:3:15: Attached scsi disk sdf
sd 0:0:3:15: Attached scsi generic sg13 type 0
```

The above output was captured just after performing a host rescan. It shows that a 300GB volume was discovered and assigned the name **/dev/sdf**.



## 4 Software iSCSI

Most major Linux distributions have been including a software iSCSI initiator for some time. Red Hat includes it in versions 4, 5, and 6 and SLES includes it in versions 9, 10 and 11. The package can be installed using the appropriate package management system (such as rpm, yum, Yast or Yast2).

Both RHEL and SLES utilize the open-iSCSI implementation of software iSCSi on the Linux platform. RHEL has included iSCSI support since version 4.2 (Oct 2005) and SLES has included open-iSCSI since version 10.1 (May 2006). iSCSI is included in several releases and has been refined over many years. It is considered to be a mature technology.

While iSCSI is considered to be a mature technology that allows organizations to economically scale into the world of enterprise storage, it has grown in complexity at both the hardware and software layers. The scope of this document is limited to the default Linux iSCSI software initiator (**open-iSCSI**). For more advanced implementations (for example, leveraging iSCSI HBAs or drivers that make use of iSCSI offload engines) consult with the associated vendor documentation and services.

For instructions on setting up an iSCSI network topology, consult the **Dell Compellent SC8000 Connectivity Guide** at <http://kc.compellent.com/Knowledge%20Center%20Documents/680-027-013.pdf>.

### 4.1 Network configuration

The system being configured requires a network port which can communicate with the iSCSI ports on the Storage Center. As a best practice, this should be a dedicated port.

The most important thing to consider when configuring an iSCSI volume is the network path. Take time to determine the sensitivity, confidentiality, security and latency needed for the iSCSI traffic. These needs will drive and define the network topology of the iSCSI architecture (for example: dedicated physical ports, VLAN usage, multipath and redundancy).

In an ideal scenario, iSCSI traffic is separated and isolated from routine network traffic by the use of dedicated ports, switches and infrastructure. If the physical topology is a constraint, it is a general best practice to separate and isolate iSCSI traffic by the use of VLAN subnetting. It is also a recommended best practice to always use iSCSI in a multipath configuration to create proper path redundancy.

If VLAN subnets are not possible, two further options that can be explored are:

- Route traffic at the network layer by defining static routes.  
and
- Route traffic at the iSCSI level via configuration.

The following directions assume that a network port has already been configured to communicate with the Storage Center iSCSI ports.



## 4.2 Configuring RHEL

The necessary tools for Red Hat servers are already contained in the *iscsi-initiator-utils* package and can be installed with yum using the command:

```
yum install iscsi-initiator-utils
```

The iSCSI software initiator consists of two main components: the daemon which runs in the background to handle connections and traffic, and the administration utility which is used to configure and modify connections. If the daemon does not start automatically during boot, it must be started before beginning the configuration.

```
service iscsi start
Starting iscsi: [OK]
chkconfig --levels 345 iscsi on
```

The next step is to discover the **iqn** names for the Storage Center ports. Storage Center SCOS versions 5.x.x and newer return all iqn names for that particular iSCSI port when applying the discovery command against the control ports.

In the example below, the iSCSI ports on the Storage Center system have the IP addresses 172.16.26.180 and 10.10.140.180 respectively.

```
iscsiadm -m discovery -t sendtargets -p 172.16.26.180:3260
172.16.26.180:3260,0 iqn.2002-03.com.compellent:5000d3100000677c
172.16.26.180:3260,0 iqn.2002-03.com.compellent:5000d3100000677e
172.16.26.180:3260,0 iqn.2002-03.com.compellent:5000d31000006780
172.16.26.180:3260,0 iqn.2002-03.com.compellent:5000d31000006782
iscsiadm -m discovery -t sendtargets -p 10.10.140.180:3260
10.10.140.180:3260,0 iqn.2002-03.com.compellent:5000d3100000677d
10.10.140.180:3260,0 iqn.2002-03.com.compellent:5000d3100000677f
10.10.140.180:3260,0 iqn.2002-03.com.compellent:5000d31000006781
10.10.140.180:3260,0 iqn.2002-03.com.compellent:5000d31000006783
```

The iSCSI daemon will save the nodes in **/var/lib/iscsi** and automatically log into them when the daemon starts. The following commands instruct the software to log into all known nodes.

```
iscsiadm -m node --login
Logging in to [iface: default, target: iqn.2002-03.com.compellent:5000d31000006782, portal: 172.16.26.180,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-03.com.compellent:5000d31000006783, portal: 10.10.140.180,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-03.com.compellent:5000d3100000677d, portal: 10.10.140.180,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-03.com.compellent:5000d3100000677f, portal: 10.10.140.180,3260] (multiple)
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d31000006782, portal: 172.16.26.180,3260] successful.
```



```
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d31000006783,
portal: 10.10.140.180,3260] successful.
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100000677d,
portal: 10.10.140.180,3260] successful.
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100000677f,
portal: 10.10.140.180,3260] successful.
[snip]
```

After running this command, a Server Object to be created on the Storage Center. After creating the Server Object and mapping a volume to the iSCSI software initiator, rescan the virtual HBA to discover new volumes presented to the host (refer to section 4.4 titled, ["Scanning for new volumes"](#)).

As long as the iSCSI daemon automatically starts during boot, the system logs in to the Storage Center iSCSI targets and discovers the iSCSI-based volumes.

## 4.3 Configuring SLES

For SLES systems, **open-iscsi** is the package that provides the iSCSI software initiator. Installing the **iscsitarget** package is optional.

The iSCSI software initiator consists of two main components: the daemon, which runs in the background and handles connections and traffic; and the administration utility, which is used to configure and modify connections. Before configuration begins, start the daemon. In addition, configure it to start automatically during boot.

```
service open-iscsi start
Starting iSCSI initiator service: done iscsiadm: no records found!
Setting up iSCSI targets: unused
chkconfig --levels 345 open-iscsi on
local:~ # iscsiadm -m discovery -t sendtargets -p 10.10.3.1
10.10.3.1:3260,0 iqn.2002-03.com.compellent:5000d3100000670c local:~ # iscsiadm
-m discovery -t sendtargets -p 10.10.3.2
10.10.3.2:3260,0 iqn.2002-03.com.compellent:5000d3100000670d
```

The system stores information about each target. After the targets have been discovered, they are accessible. This creates the virtual HBAs as well as any device files for volumes mapped at login time.

```
iscsiadm -m node --login
```

The last step is to configure the system to automatically login to the targets during a boot when the iSCSI software initiator starts.

```
iscsiadm -m node --op=update --name=node.startup --value=automatic
```



## 4.4 Scanning for new volumes

The process for scanning the iSCSI software initiator for new volumes is identical to that used for scanning the Fibre Channel hostX devices as discussed in section 2.1.

## 4.5 /etc/fstab configuration

Since iSCSI is dependent on a running network, volumes added to **/etc/fstab** need to be designated as network-dependent. In other words, do not attempt to mount an iSCSI volume until the network layer services have completed the startup and the network is up. The example below demonstrates how to create this network dependency to the iSCSI mount using the **\_netdev** mount option in the **/etc/fstab** file.

```
LABEL=iscsiVOL /mnt/iscsi ext3 _netdev 0 0
```

## 4.6 iSCSI timeout values

In a single path environment, configure the iSCSI daemon to wait and queue I/O for a sufficient amount of time to accommodate a proper failure recovery. For example, a SAN fabric failure or Storage Center failover event can take between 30 and 60 seconds to complete. Therefore, it is a recommended best practice to configure the iSCSI software initiator to queue I/O for 60 seconds before starting to fail I/O requests.

When iSCSI is used in a multipath environment, configure the iSCSI daemon to fail a path in about 5 seconds. This minimizes the latency of waiting for a single path to recover. Since I/O is managed at the higher dm-multipath layer, it will automatically resubmit any I/O requests to alternating active iSCSI routes. If all routes are down, the dm-multipath layer will queue the I/O until a route becomes available. This multi-layer approach allows an environment to sustain failures at both the network and storage layers.

The following configuration settings directly affect iSCSI connection timeouts.

To control how often a **NoOp-Out** request is sent to each target, configure the following parameter.

```
node.conn[0].timeo.noop_out_interval = X
```

**X** is measured in seconds, and the default value is 10.

To control the timeout value for the NoOp-Out request, configure the following parameter.

```
node.conn[0].timeo.noop_out_timeout = X
```

**X** is measured in seconds, and the default value is 15.

The next iSCSI timer that needs to be modified is:

```
node.session.timeo.replacement_timeout = X
```

**X** is measured in seconds, and the default value is 15.



The **replacement\_timeout** command controls the wait time of a session re-establishment before failing the pending SCSI commands. This includes commands that the SCSI layer error handler is passing up to a higher level (such as multipath) or to an application if multipath is not active.

The **NoOp-Out** section dictates that if a network problem is detected, the running commands are failed immediately. The exception is if the SCSI layer error handler is running. To check if the SCSI error handler is running, run the following command.

```
iscsiadm -m session -P 3
Host Number: X State: Recovery
```

When the SCSI error handler is running, commands will not be failed until the seconds of the **node.session.timeo.replacement\_timeout** parameter are modified.

To modify the timer that starts the SCSI error handler, run the following command.

```
echo X > /sys/block/sdX/device/timeout
```

*X* is measured in seconds. Depending on the Linux distribution, this can also be achieved by modifying the respective **udev** rule.

To modify the **udev** rule, edit the **/etc/udev/rules.d/60-raw.rules** file and append the following lines.

```
ACTION=="add", SUBSYSTEM=="scsi" , SYSFS{type}=="0|7|14", \
RUN+="/bin/sh -c 'echo 60 > /sys$DEVPATH/timeout'"
```

## 4.7 Multipath timeout values

The following line in **/etc/multipath.conf** will instruct **dm-multipath** to queue I/O in the event that all paths are down. This line configures multipath to wait for the Storage Center to recover in case of a controller failover event.

```
features "1 queue_if_no_path"
```

Timeout and other connection settings are statically created during the discovery step and written to the configuration files in **/var/lib/iscsi/\***.

There is not a specific timeout value that is appropriate for every environment. In a multipath Fibre Channel environment, it is recommended to set timeout values on the FC HBA to five seconds. However, take additional caution when determining the appropriate value to use in an iSCSI configuration. Since iSCSI is often used on a shared network, it is extremely important to avoid inadvertent non-iSCSI network traffic from interfering with the iSCSI storage traffic.

It is important to take into consideration the different variables of an environment configuration and thoroughly test all perceivable failover scenarios (such as switch, port, fabric and controller) before deploying them into a production environment.



## 5 Server configuration

This section discusses configuring the multiple aspects of the I/O stack on a Linux host. How the stack behaves and performs can be precisely configured to the needs of the environment in which the Linux hosts operate. The configuration aspects include tuning the HBA port retry count, the queue depth, the SCSI device timeout values and many more. “Managing modprobe” provides a starting point and outlines a few configuration parameters which should be taken into account as they are adapted for individual environment use.

### 5.1 Managing modprobe

The **modprobe** facility of Linux provides a means to manage and configure the operating parameters of installed HBAs (QLogic, Emulex or otherwise) on a Linux host. As with managing most aspects of UNIX and/or Linux, the modprobe facility is configured by editing text files as outlined below. The actual configuration syntax for QLogic and Emulex hardware is discussed in sections 5.4, [“In single-path environments”](#) and 5.5, [“In multipath environments”](#) respectively.

#### 5.1.1 Red Hat Linux 6, SLES 11/10 and newer

The QLogic and Emulex HBAs are managed by configuration syntax that is written within individual files, located in the **/etc/modprobe.d** directory.

- For Qlogic:

```
/etc/modprobe.d/qla2xxx.conf
```

- For Emulex:

```
/etc/modprobe.d/lpfc.conf
```

#### 5.1.2 Red Hat Linux 5, SLES 9 and older

The QLogic and Emulex HBAs are managed by appending the required configuration syntax to the end of the **/etc/modprobe.conf** file.

#### 5.1.3 Reloading modprobe and RAM disk (mkinitrd)

After configuration changes are made, reload the modprobe facility for the new or updated configuration to take effect.

For local boot systems, it is recommended to unmount all SAN volumes and then reload the module. The module should be unloaded from memory before it is reloaded again as shown below.

```
modprobe -r qla2xxx
modprobe qla2xxx
```

Replace **qla2xxx** with **lpfc** if working with Emulex hardware.



Afterwards, SAN volumes can be remounted.

For boot from SAN systems, the RAM disk also needs to be rebuilt so that the new configurations are incorporated during boot time. The procedure below demonstrates rebuilding the **initrd** file. This process overwrites the same file at its existing location. It is recommended to copy and backup the existing **initrd** file before applying this procedure.

```
mkinitrd -f -v /boot/initrd-<kernel version>.img <kernel version>
```

Observe the output from the command and make sure that the **Adding module** line for the applicable module has the correct options.

```
mkinitrd -f -v /boot/initrd $(uname -r) [snip]
Adding module qla2xxx with options qlport_down_retry=60 [snip]
```

Restart the system to ensure that the GRUB entry in **/etc/grub.conf** points to the correct **initrd** file.

### 5.1.4 Verifying parameters

To verify that the configuration changes have taken effect, apply the following commands.

- For Qlogic:

```
cat /sys/module/qla2xxx/parameters/qlport_down_retry
60
```

- For Emulex:

```
cat /sys/class/scsi_host/host0/lpfc_devloss_tmo
60
```

## 5.2 SCSI device timeout configuration

The SCSI device timeout is configured to 60 seconds by default. Do not change this value unless instructed or recommended to do so by specific vendor requirements. Verify the setting with the following command.

```
cat /sys/block/sdX/device/timeout
60
```

## 5.3 Queue depth configuration

There are two places to set the queue depth for Fibre Channel HBAs. One place it can be configured is in the BIOS for the HBA. This value can be modified during boot time or by using the tools provided by the HBA vendor. The queue depth can also be configured using the HBA configuration files that are managed by the **modprobe** facility at the OS level. If these two numbers are different, the lower of the two numbers takes precedence. It is recommended to configure the BIOS setting to a high value, and then manage the HBA queue depth with its respective OS level configuration file. The default queue depth is set to 32.



However, a queue depth of 128 is a recommended starting value. Assess, test and adjust the value accordingly to meet individual environment needs.

- For Qlogic:

```
options qla2xxx ql2xmaxqdepth=<value>
```

- For Emulex:

```
options lpfc lpfc_lun_queue_depth=<value> lpfc_hba_queue_depth=<value>
```

## 5.4 In single-path environments

During a Storage Center failover event, Legacy port mode behavior causes the WWPN of an active port on the failed controller to disappear from the fabric momentarily before relocating to a reserve port on an active controller. In Virtual port mode for path failover, the WWPN on any active NPIV port will relocate to another active NPIV port of the same fault domain. For controller failover, the WWPN relocates to another active NPIV port of the same fault domain on the other active controller. In either failover scenario, the Storage Center may take anywhere from five to 60 seconds to propagate these changes through the SAN fabric.

In order to mitigate any I/O disruption in single-path connectivity scenarios, it is recommended to instruct the HBA level code to wait for up to 60 seconds (from its default of 30 seconds) before marking a port as down or failed. This would allow the Storage Center sufficient time to relocate the WWPN of the failed port to an active port and propagate the changes through the SAN fabric. The configuration syntax for a single path environment is outlined in section 5.4.1.

### 5.4.1 PortDown timeout

These settings dictate how long a Linux system waits before destroying a connection after losing its connectivity with the port. The following should be configured inside the **qla2xxx.conf**, **lpfc.conf** or **modprobe.conf** files accordingly.

- For Qlogic:

```
options qla2xxx qlport_down_retry=60
```

- For Emulex:

```
options lpfc lpfc_devloss_tmo=60
```



## 5.5 In multipath environments

During a Storage Center failover event, Legacy port mode behavior causes the WWPN of any active port on the failed controller to disappear from the fabric momentarily before relocating to a reserve port on an active controller. In Virtual port mode for path failover, the WWPN on any active NPIV port will relocate to another active NPIV port of the same fault domain. For controller failover, the WWPN relocates to another active NPIV port of the same fault domain on the other active controller. In either failover scenario, the Storage Center may take anywhere from five to 60 seconds to propagate these changes through a fabric.

In order to mitigate I/O disruption in multipath connectivity scenarios; it is recommended to instruct the HBA level code to wait up to five seconds (instead of the default 30 seconds) before marking a port as down/failed. This minimizes I/O latency to the I/O or application stack above it by quickly relocating I/O requests to alternate and active HBA paths through the SAN fabric. If all of the paths are down, the **dm-multipath** module (further up the I/O stack) will start queuing I/O until one or more paths have recovered to an active state. This allows the Storage Center sufficient time to relocate the WWPN of the failed port to an active port and propagate the changes through the SAN fabric. The configuration syntax for a multipath environment is outlined in the section 0.

In SAS connected environments, paths from both controllers are presented to the connected host (Active/Optimized and Standby), however only the Active/Optimized path is used for all active I/O at any one time. When the Active/Optimized path becomes unavailable, Storage Center will determine which one of the remaining Standby paths will assume the role of the Active/Optimized path and continue to stream active I/O to the new Active/Optimized path.

### 5.5.1 PortDown timeout

Configure the following inside the **qla2xxx.conf**, **lpfc.conf** or **modprobe.conf** files accordingly.

- For Qlogic:

```
options qla2xxx qlport_down_retry=5
```

- For Emulex:

```
options lpfc lpfc_devloss_tmo=5
```

### 5.5.2 Multipath a volume

Even if a volume is initially mapped in single-path mode, use multipath whenever possible.

The first step in configuring multipath for a volume is to create the necessary mappings. It is possible to configure a volume as multipath with only one path existing. However, in order to achieve the benefits, it is necessary to have at least two paths.

In the example below, the server has two Fibre Channel ports. The Storage Center has two front end ports on each controller. They are zoned in two separate virtual SANs to create a redundant SAN fabric.



Start by creating a new Server Object in the server tree of the Storage Center GUI. The wizard displays prompts to associate the server ports to the new Server Object as shown below. Click **Continue** once the HBAs have been selected.

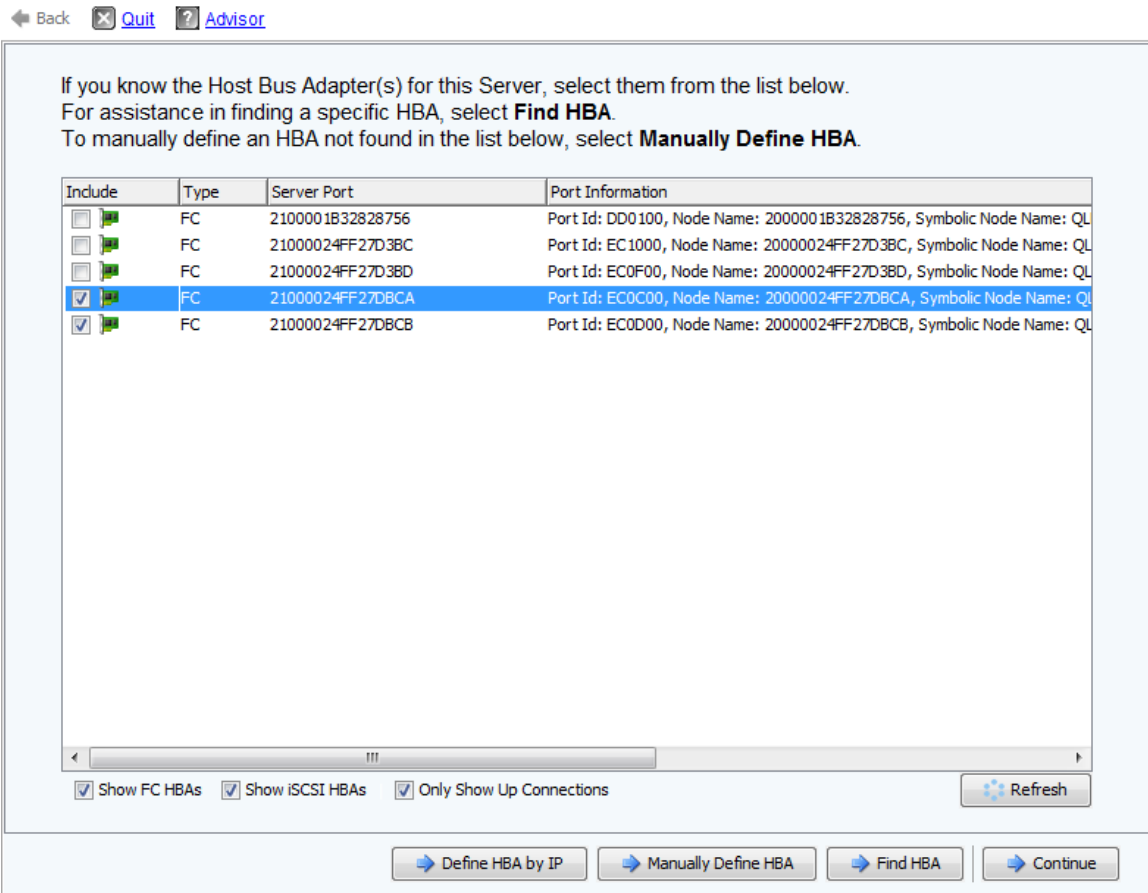


Figure 3 Select the server ports to associate to the Server Object

Place the new Server Object in the desired server tree folder, name it and define the operating system. Click **Continue** to proceed.



[Back](#)
[Quit](#)
[Advisor](#)

Folder:

- Servers
  - AIX
  - HP-UX
  - LinuxRHEL**
  - LinuxSuSE
  - Oracle
  - Solaris
  - Symantec

[Create a New Folder](#)

Name:

Operating System:

Notes:

[Continue](#)

Figure 4 Locate, name and define the new Server Object

Finally, click **Create Now** to complete the process. The new Server Object and mapped volumes will be presented to the Linux system in multipath mode.

On the Linux system, scan and discover newly presented volumes as detailed in section 2.1, [“Scanning for new volumes”](#).

Discovered volumes can be correlated to their respective storage center devices as detailed in sections 3.1, [“The lsscsi command”](#) and 3.2, [“The scsi\\_id command”](#) respectively.

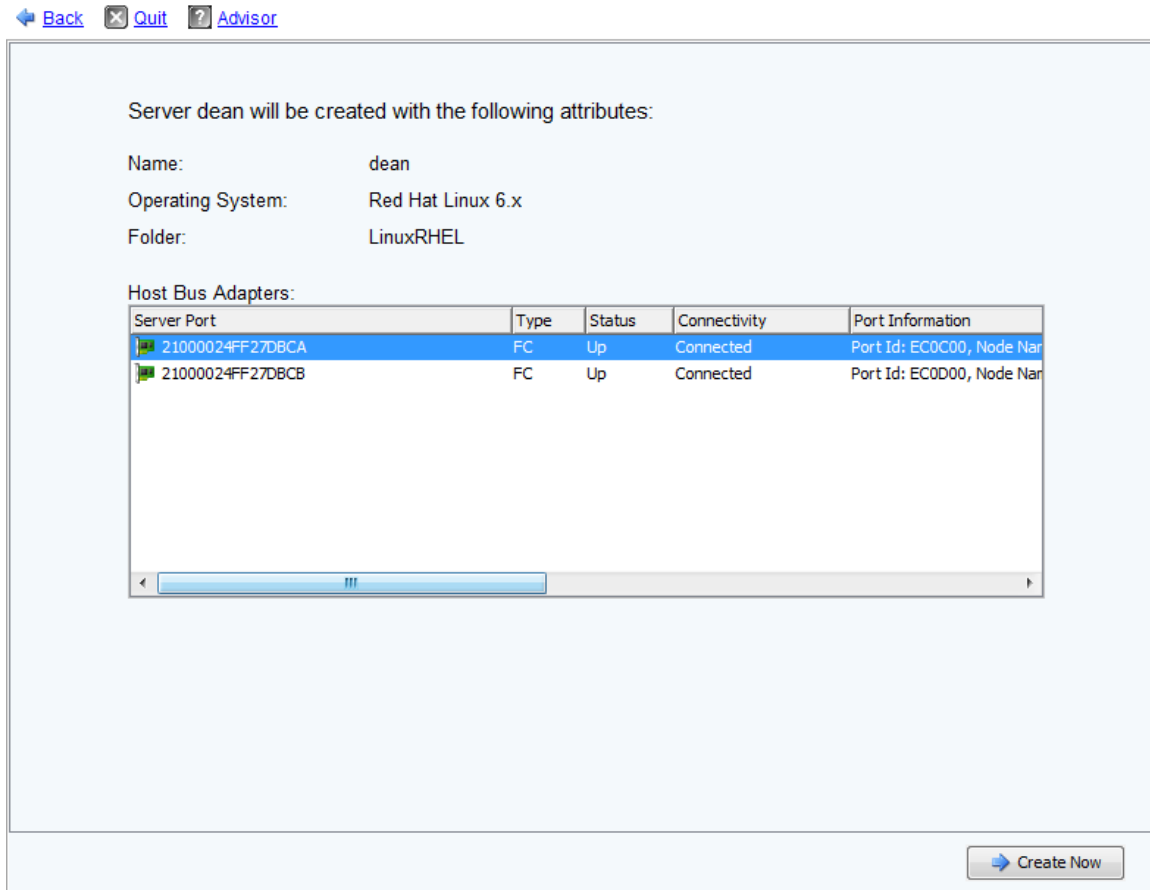


Figure 5 Click **Create Now**

The following multipath command may be issued to display all discovered multipath volumes.

```
multipath -ll
DEMO-VOL (36000d3100000690000000000000001483) dm-4 COMPELNT,Compellent Vol
size=20G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
 |- 6:0:5:200 sdb 8:16 active ready running
 `-- 7:0:7:200 sdd 8:48 active ready running
BOOT-VOL (36000d3100000670000000000000000a68) dm-0 COMPELNT,Compellent Vol
size=64G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
 |- 7:0:5:0 sda 8:0 active ready running
 `-- 6:0:7:0 sdc 8:32 active ready running
LUN02 (36000d31000006900000000000000014ce) dm-5 COMPELNT,Compellent Vol
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
 |- 6:0:4:2 sde 8:64 active ready running
 `-- 7:0:6:2 sdh 8:112 active ready running
LUN01 (36000d31000006900000000000000014cd) dm-6 COMPELNT,Compellent Vol
```



```

size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
 |- 6:0:4:1 sdf 8:80 active ready running
 `-- 7:0:6:1 sdi 8:128 active ready running
LUN00 (36000d31000006900000000000000014cc) dm-7 COMPELNT,Compellent Vol
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
 |- 6:0:5:1 sdg 8:96 active ready running
 `-- 7:0:7:1 sdj 8:144 active ready running

```

### 5.5.3 Multipath aliases

Linux multipath will automatically generate a new name for each multipath device it discovers and manages. Unlike the **/dev/sdX** device names assigned to block devices, multipath names are persistent across reboots and reconfiguration. This means that multipath names are consistent and safe for use in scripting, mount commands, **/etc/fstab** and more. Additionally, multipath names (which by default appear as **/dev/mapper/mpathX**) can be assigned aliases by way of configuration syntax inside the **/etc/multipath.conf** file. Designating aliases is recommended and extremely useful when associating multipath device names with more descriptive labels (for example, business function or usage).

Assigning an alias to a multipath volume is accomplished inside the multipath clause in the **/etc/multipath.conf** file as shown below. The **wwid** key value represents the wwid value of the SAN volume and the **alias** key value represents a user-defined description of the volume. Once a volume name is defined, reference by that alias (**/dev/mapper/<alias name>**) for all purposes.

```

multipaths {
 multipath {
 wwid "36000d3100003600000000000000000837"
 alias "Oracle_Backup_01"
 [snip]
 }
}

```

After defining aliases or updating any constructs inside **/etc/multipath.conf**, the multipathd daemon should be reloaded to reflect any changes.

```
service multipathd reload
```



## 5.5.4 Storage Center device definition

The Storage Center device definitions do not exist in the kernel in the Red Hat Linux versions 5.4 and older. Add the following syntax to the **/etc/multipath.conf** file to compensate and allow the Linux kernel to properly identify and manage Storage Center volumes.

```
devices {
 device {
 vendor "COMPELNT"
 product "Compellent Vol"
 path_checker tur
 no_path_retry queue
 }
}
```

With Red Hat Linux versions newer than 5.4, the Storage Center device definitions are already incorporated into the kernel. Use **/etc/multipath.conf** within its native syntax constraints as shown in the example below. This **/etc/multipath.conf** configuration applies only in FC and iSCSI implementations; the SAS **/etc/multipath.conf** configuration is discussed in [Section 5.6.2](#) and should be used instead in SAS implementations.

```
cat /etc/multipath.conf
multipath.conf written by anaconda

defaults {
 user_friendly_names yes
}
blacklist {
 devnode "^ (ram|raw|loop|fd|md|dm-|sr|scd|st) [0-9] *"
 devnode "^hd[a-z]"
 devnode "^dcssblk[0-9] *"
 device {
 vendor "DGC"
 product "LUNZ"
 }
 device {
 vendor "IBM"
 product "S/390.*"
 }
 # don't count normal SATA devices as multipaths
 device {
 vendor "ATA"
 }
 # don't count 3ware devices as multipaths
 device {
 vendor "3ware"
 }
 device {
```



```

 vendor "AMCC"
 }
 # nor highpoint devices
 device {
 vendor "HPT"
 }
 wwid "20080519"
 wwid "20080519"
 device {
 vendor iDRAC
 product Virtual_CD
 }
 device {
 vendor PLDS
 product DVD-ROM_DS-8D3SH
 }
 #wwid "*"
}
blacklist_exceptions {
 device {
 vendor "COMPELNT"
 product "Compellent Vol"
 }
}
}
multipaths {
 multipath {
 alias BOOT-VOL
 uid 0
 gid 0
 wwid "36000d31000006700000000000000000a68"
 mode 0600
 }
 multipath {
 alias DEMO-VOL
 uid 0
 gid 0
 wwid "36000d310000069000000000000000001483"
 mode 0600
 }
 multipath {
 alias LUN00
 uid 0
 gid 0
 wwid "36000d3100000690000000000000000014cc"
 mode 0600
 }
 multipath {

```



```

 alias LUN01
 uid 0
 gid 0
 wwid "36000d31000006900000000000000014cd"
 mode 0600
 }
 multipath {
 alias LUN02
 uid 0
 gid 0
 wwid "36000d31000006900000000000000014ce"
 mode 0600
 }
}

```

## 5.6 Serial Attached SCSI

The Dell SCv2000 series product line offers Serial Attached SCSI (SAS) front end connectivity and coincides with the launch of SCOS 6.6.x. The Dell SCv2000 series supports the use of Dell 12Gbps SAS HBAs on the target hosts.

SAS connectivity to a Linux host requires specific configuration schema in the `/etc/multipath.conf` file.

### 5.6.1 SAS drivers

SAS drivers are preloaded into certain Linux kernels (RHEL 6.5/newer and RHEL 7x). The existence of the SAS drivers can be validated with the following commands. As a best practice, validate the driver versions with the Dell Storage Compatibility Matrix and use the latest supported driver as indicated.

```

lsmod | grep sas
mpt3sas 188001 4
scsi_transport_sas 35588 1 mpt3sas
raid_class 4388 1 mpt3sas
megaraid_sas 96205 2

modinfo mpt3sas | grep description
description: LSI MPT Fusion SAS 3.0 Device Driver

```

### 5.6.2 SAS `/etc/multipath.conf`

Add the following device schema to the `/etc/multipath.conf` file and used exclusively for SAS connected Linux hosts. This schema defines the configuration parameters for all devices identified by `vendor="COMPELNT"` and `product="Compellent Vol"`. This schema is typically added after the defaults schema and before the blacklist\_exceptions schema.

```

devices {
 device {

```



```

 vendor COMPELLNT
 product "Compellent Vol"
 path_checker tur
 prio alua
 path_selector "service-time 0"
 path_grouping_policy group_by_prio
 no_path_retry 24
 hardware_handler "1 alua"
 failback immediate
 rr_weight priorities
 }
}

```

### 5.6.3 FC/iSCSI & SAS

The use of a merged FC/iSCSI & SAS connected environment on the same Linux host is not validated nor supported. Maintain a SAS connected Linux host that is separate from other Linux hosts using FC/iSCSI connectivity.

The SAS specific multipath.conf configuration schema can be merged with a FC/iSCSI-based /etc/multipath.conf file. Note that this merged configuration is not supported by Dell.

### 5.6.4 Identify SAS devices on Linux

This sample script parses the Linux /sys filesystem and displays the contents of the files located and identified as host\_sas\_address. The contents of this file represents the device name of the installed SAS HBA. This script is provided AS-IS without any stated warranty or support of any kind.

In the instance below, two SAS HBA cards are identified with their device names shown accordingly. This script works for both RHEL 6x and RHEL 7x Linux hosts.

```

for i in `find /sys/devices -name host_sas_address`; do echo "=== $i"; cat $i;
echo; done
===
/sys/devices/pci0000:40/0000:40:01.0/0000:41:00.0/host1/scsi_host/host1/host_sas
_address
0x544a842007902800

===
/sys/devices/pci0000:40/0000:40:03.0/0000:42:00.0/host2/scsi_host/host2/host_sas
_address
0x544a84200792ce00

```

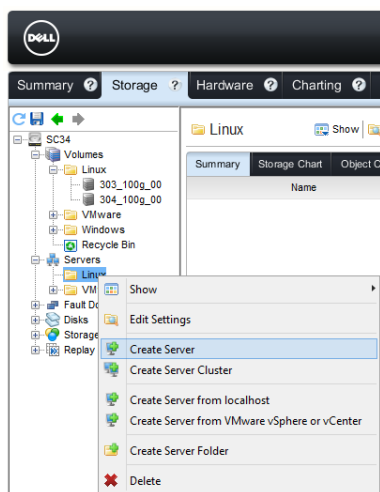


## 5.6.5 Identify SAS devices on Dell SCv2000

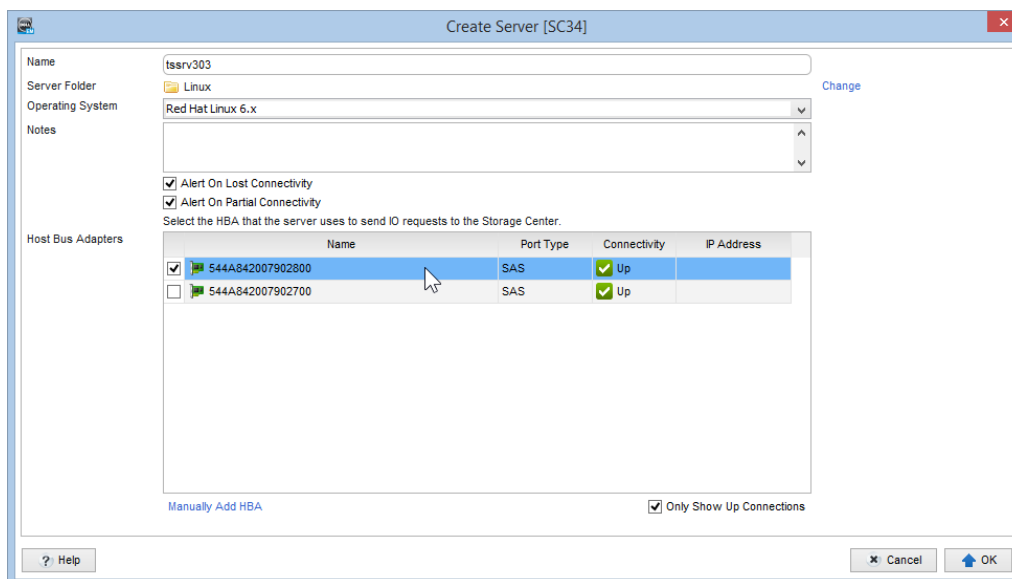
SAS devices are exposed to the Dell SCv2000 via their singular device names instead of their multiple SAS world-wide port names (WWPNs).

The following sequence of screenshots outlines the procedure to create a new server object on a Dell SCv2000 storage array.

1. Right click in the **Servers** folder tree and select **Create Server**.



2. Name the new server object, select the proper Operating System (Red Hat Linux 6.x or 7.x), select the identified SAS device name corresponding to the Linux host, and then click **OK**.



- The new server object (tssrv303) is created in the Servers folder tree, with the Connectivity pane displaying that the Linux host has established connectivity to both bottom and top controllers of the Storage Center array.

The screenshot shows the Dell Storage Center web interface. The left sidebar displays a tree view with folders for Volumes, Servers, Fault Domains, Disks, Storage Types, and Replay Profiles. The 'Servers' folder is expanded, showing a list of servers including 'tssrv303'. The main pane displays the 'Connectivity' tab for server 'tssrv303'. The 'Index' section shows '21' and 'Server has not used any disk space on the Storage Center'. The 'Server HBAs' table shows one HBA with ID '544A842007902800', Port Type 'SAS', and Connectivity 'Up'. The 'Mappings' section shows '544A842007902800' and 'SAS Domain 1'. The 'Connectivity' table shows two controllers: 'Bottom Controller' and 'Top Controller', both with 'Up' status.

| Name             | Port Type | Connectivity |
|------------------|-----------|--------------|
| 544A842007902800 | SAS       | Up           |

| Controller        | Controller Port           | Path Status | Slot | Slot Port | Fault Domains |
|-------------------|---------------------------|-------------|------|-----------|---------------|
| Bottom Controller | FE 5000D31000FEB315 (1-1) | Up          | 1    | 1         |               |
| Top Controller    | FE 5000D31000FEB308 (1-1) | Up          | 1    | 1         |               |

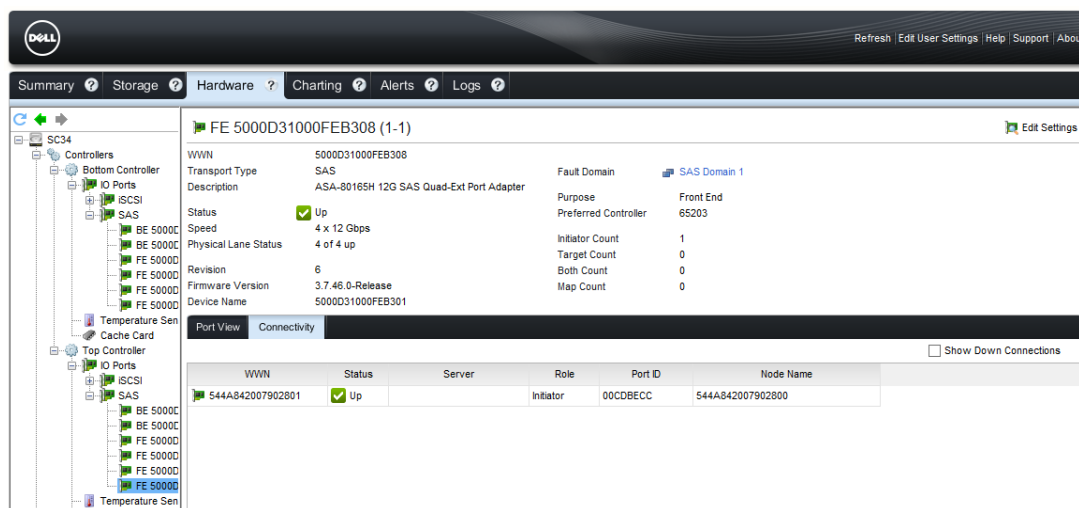
- Select the SAS controller ending \*FEB315 in the Hardware pane to display the actual SAS WWPN ending \*2800 in the Connectivity pane.

The screenshot shows the Dell Storage Center web interface. The left sidebar displays a tree view with folders for Controllers, IO Ports, SAS, Temperature Sen, Cache Card, Enclosures, and Enclosure - 1. The 'Controllers' folder is expanded, showing a list of controllers including 'FE 5000D31000FEB315 (1-1)'. The main pane displays the 'Connectivity' tab for controller 'FE 5000D31000FEB315 (1-1)'. The 'WWN' section shows '5000D31000FEB315'. The 'Transport Type' is 'SAS'. The 'Description' is 'ASA-80165H 12G SAS Quad-Ext Port Adapter'. The 'Status' is 'Up'. The 'Speed' is '4 x 12 Gbps'. The 'Physical Lane Status' is '4 of 4 up'. The 'Revision' is '6'. The 'Firmware Version' is '3.7.46.0-Release'. The 'Device Name' is '5000D31000FEB302'. The 'Fault Domain' is 'SAS Domain 1'. The 'Purpose' is 'Front End'. The 'Preferred Controller' is '65204'. The 'Initiator Count' is '1'. The 'Target Count' is '0'. The 'Both Count' is '0'. The 'Map Count' is '0'. The 'Port View' section shows a table with columns: WWN, Status, Server, Role, Port ID, and Node Name. The table shows one entry: '544A842007902800', 'Up', 'tssrv303', 'Initiator', '001699B8', and '544A842007902800'.

| WWN              | Status | Server   | Role      | Port ID  | Node Name        |
|------------------|--------|----------|-----------|----------|------------------|
| 544A842007902800 | Up     | tssrv303 | Initiator | 001699B8 | 544A842007902800 |



5. Select the SAS controller ending \*FEB308 in the Hardware pane to display the actual SAS WWPN ending \*2801 in the Connectivity pane.



## 5.6.6 Configured multipath

A properly configured SAS volume will return the following “multipath -ll” output. This Storage Center volume is discovered as a multipath ALUA-capable volume, where each path is capable of I/O. The path represented by prio=50 (Active/Optimized path) is used for all active I/O requests. The path represented by prio=1 (Standby path) is a highly available, redundant path and is used when the Active/Optimized path becomes unavailable.

```
multipath -ll
Compelnt_0016 (36000d31000feb30000000000000000016) dm-3 COMPELNT,Compellent Vol
size=100G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| `-- 1:0:0:1 sdb 8:16 active ready running
`+- policy='service-time 0' prio=1 status=enabled
 `-- 1:0:1:1 sdc 8:32 active ready running
Compelnt_001a (36000d31000feb3000000000000000001a) dm-4 COMPELNT,Compellent Vol
size=100G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| `-- 1:0:1:2 sdd 8:48 active ready running
`+- policy='service-time 0' prio=1 status=enabled
 `-- 1:0:2:2 sde 8:64 active ready running
```



### 5.6.7 SAS queue depth

Supported SAS HBAs default to a queue depth of 254 per volume. This value should be left at its factory default value unless specifically dictated by an application layer configuration requirement.

```
lsscsi -L | egrep 'COMPELNT|depth'
[1:0:0:1] disk COMPELNT Compellent Vol 0606 /dev/sdb
 queue_depth=254
[1:0:1:1] disk COMPELNT Compellent Vol 0606 /dev/sdc
 queue_depth=254
```

### 5.6.8 Boot from SAS

Boot from SAS functionality is not validated nor supported in use with Dell 12Gbps SAS HBAs on Linux.



## 6 Performance tuning

This section provides general information and guidance pertaining to some of the more common performance tuning options and variables available to Linux, particularly with RHEL versions 5.9 thru 6.4. This information is not intended to be all-encompassing and the values used should not be considered final. The intent of this section is to provide a starting point from which Linux and storage administrators can fine-tune their Linux installation to achieve optimal performance.

Prior to making any changes to the following parameters, a good understanding of the current environment workload should be established. There are numerous methods by which this can be accomplished including the perception of the system or storage administrators based on day-to-day experience with supporting the environment. The Dell Performance Analysis Collection Kit (DPACK) is a free toolkit which can be obtained by sending an email to the address below.

Email: [DPACK\\_Support@Dell.com](mailto:DPACK_Support@Dell.com)

Some general guidelines to keep in mind for performance tuning with Linux are:

- Performance tuning is as much an art as it is a science. Since there are a number of variables which impact performance (I/O in particular), there are no specific values that can be recommended for every environment. Begin with a few variables and add more variables or layers as system is tuned. For example, start with single path, tune and then add multipath.
- Make one change at a time and then test, measure and assess the impact on performance with a performance monitoring tool before making any subsequent changes.
- It is considered a best practice to make sure the original settings are recorded so the changes can be reverted to a known state if needed.
- Apply system tuning principles (e.g. failover) in a non-production environment first (where able) and validate the changes with as many environmental conditions as possible before propagating these changes into production environments.
- If performance needs are being met with the current configuration settings, it is generally a best practice to leave the settings alone to avoid introducing changes that may make the system less stable.
- An understanding of the differences between block and file level data should be established in order to effectively target the tunable settings for the most effective impact on performance. Although the Storage Center array is a block-based storage device, the support for the iSCSI transport mechanism introduces performance considerations that are typically associated with network and file level tuning.

When validating whether a change is having an impact on performance, leverage the charting feature of the Dell Enterprise Manager to track the performance. In addition, be sure to make singular changes between iterations in order to better track what variables have the most impact (positive or negative) on I/O performance.



## 6.1 Leveraging the use of multiple volumes

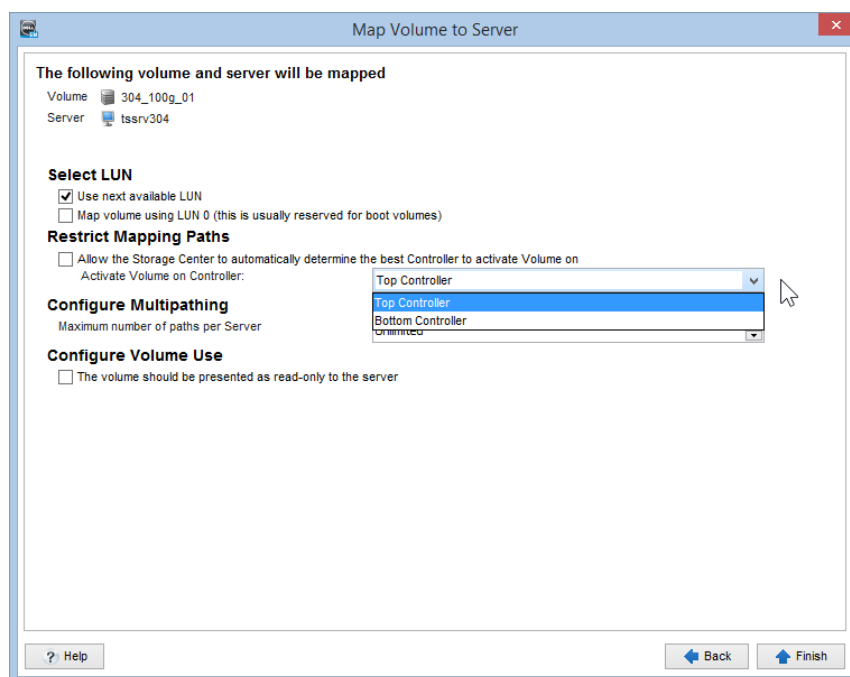
A volume can only be active on one Storage Center controller at a time. Therefore, when possible, spread volumes evenly across both Storage Center controllers to most effectively leverage dual I/O processing. A larger number of smaller-sized volumes will often result in better performance than fewer larger-sized volumes. From a Linux perspective, having multiple target volumes can result in performance improvements by leveraging the kernel to process I/O in parallel to addressing multiple paths, SCSI devices and others.

In SAS connected environments, paths from both controllers are presented to the connected host (Active/Optimized and Standby), however only the Active/Optimized path is used for all active I/O at any one time. The methods described above can be applied to achieve greater I/O threading and throughput across multiple controllers by using a different Active/Optimized path for each volume. This is accomplished by explicitly pinning volumes to a different controller when mapping these volumes to the server object. This feature is accessible via the Advanced Options on the mapping dialog as shown below.

Click on the Advanced Options link.



Uncheck the Restrict Mapping Paths checkbox, then select which controller in which to pin the volume for mapping to the server object.



## 6.2 Understanding HBA queue depth

Queue depth refers to the number of pending I/O requests. Modifying this value can lead to an improvement in I/O performance in some workloads. Generally, increasing queue depth can increase throughput, but caution should be taken as increasing this value can also lead to higher latency. Different applications may benefit from increasing this value, such as environments in which the bulk of I/O is small reads/writes. In environments defined by lower IOPS requirements but needing higher throughput, this may be achieved by lowering this queue depth setting until optimal levels of performance are achieved.

This value can be changed in the HBA firmware or in the Linux kernel module for the HBA. Keep in mind that if the two settings have different values, the lower value takes precedence. Therefore, one good strategy to consider would be setting the HBA firmware to a high number and then tune the value downward from within the Linux kernel module.



Consult section 5, [“Server configuration”](#) for details on modifying this value for the particular HBA model being used.

## 6.3 Linux SCSI device queue variables

There are several Linux SCSI device queue settings that can be tuned to improve performance. The most common ones are listed below, with a brief explanation of what each parameter does with regard to I/O. These values are found in the `/sys/block/<device>/queue` directory and should be modified for each device in the volume targeted for performance modification.

### 6.3.1 Kernel IO scheduler

The `/sys/block/<device>/queue/scheduler` parameter and its contents define the I/O scheduler in use by the Linux kernel for SCSI (sd) devices. Some application vendors (such as Oracle) provide specific recommendations for which I/O scheduler to use in order to achieve optimal performance with the application platform. By default on RHEL 6/5 this is set to **cfq** as denoted by the `[ ]` brackets within the file. This parameter can be dynamically changed by performing the following command.

```
cat /sys/block/sda/queue/scheduler noop anticipatory deadline [cfq]
echo deadline > /sys/block/sda/queue/scheduler
cat /sys/block/sda/queue/scheduler noop anticipatory [deadline] cfq
```

The above command changes the I/O scheduler for the **sda** SCSI device to use the **deadline** option instead of **cfq**. This command is applied for the current running instance of the OS. A script could be used to make this change persistent on all required SCSI (sd) devices (on a per device basis) during boot time. Alternatively, this change can also be applied system wide during boot time by appending the **elevator=** key value option to the end of the kernel string inside of the `/etc/grub.conf` boot configuration file as shown below.

```
kernel /vmlinuz-2.6.18-238.el5 ro root=/dev/sda3 quiet elevator=deadline
```

**Note:** In multipath and LVM configurations, this modification should be made to each device used by the device-mapper subsystem.

### 6.3.2 read\_ahead\_kb

This parameter is used when the kernel detects it is sequentially reading from a block device and defines how many kilobytes of I/O the Linux kernel will read. Modifying this value can have a noticeable effect on performance in heavy sequential read workloads. By default with RHEL 6/5, this value is set to 128. Increasing this to a larger size may result in higher read throughput performance.

### 6.3.3 nr\_requests

This value is used by the Linux kernel to set the depth of the request queue and is often used in conjunction with changes to the queue depth configuration of the HBA. With the **cfq** I/O scheduler, this is set to 128 by default. Increasing this value sets the I/O subsystem to a larger threshold to which it will continue scheduling requests. This keeps the I/O subsystem moving in one direction longer, which can



result in more efficient handling of disk I/O. It is a best practice starting point to increase this value to 1024, assess and adjust accordingly per the performance results observed and achieved.

### 6.3.4 `rr_min_io`

When taking advantage of a multipath configuration where multiple physical paths can be leveraged to perform I/O operations to a multipath device, the `rr_min_io` parameter can be modified to optimize the I/O subsystem. The `rr_min_io` specifies the number of I/O requests to route to a path before switching paths in the current path group. The `rr_min_io` parameter is applicable to kernel versions 2.6.31 and older.

The default value of `rr_min_io` is 1000 and is generally agreed to be much too high. As a general rule of thumb, set the value two times higher than the queue depth and then assess the performance. The goal of modifying this parameter is to try and create an I/O flow which most efficiently fills up the I/O buckets in equal proportions as it is passes through the Linux I/O subsystem.

With kernel versions newer than 2.6.31, the `rr_min_io` parameter has been superseded by `rr_min_io_rq` instead. This new parameter defaults to one (meaning I/O is load balanced and issued to each path in the current path group in a round robin fashion). When this parameter is used in concurrence with the `rr_weight` parameter (which defaults to *uniform*), there is no further nor recommended need to tune this parameter from its default value.

This value is modified by making changes to the `/etc/multipath.conf` file in the **defaults** clause as shown below.

```
#defaults {
udev_dir /dev
polling_interval 10
selector "round-robin 0"
path_grouping_policy multibus
[snip]
rr_min_io_rq 1
rr_weight uniform
[snip]
#}
```

## 6.4 iSCSI considerations

Tuning performance for iSCSI is as much an effort in Ethernet network tuning as it is block-level tuning. Many of the common Ethernet kernel tunable parameters should be experimented with in order to determine which settings provide the highest performance gain with iSCSI. The use of jumbo frames can simply and often lead to improved iSCSI performance when used with 1Gb/10Gb Ethernet. As with Fibre Channel, changes should be made individually, incrementally and evaluated against multiple workload types expected in the environment in order to fully understand the effects on overall performance.



In other words, tuning performance for iSCSI is often more time consuming as one must consider the block-level subsystem tuning as well as network (Ethernet) tuning. A solid understanding of the various Linux subsystem layers involved is necessary to effectively tune the system.

Kernel parameters that can be tuned for performance are found in the **/proc/sys/net/core** and **/proc/sys/net/ipv4 kernel** parameters. Once optimal values are determined, these can be permanently set in the **/etc/sysctl.conf** file.

Like most other modern OS platforms, Linux can do a good job of auto-tuning TCP buffers. However, by default, some of the settings are set conservatively low. Experimenting with the following kernel parameters can lead to improved network performance, which would then subsequently improve iSCSI performance.

- TCP Max Buffer Sizes:
  - net.core.rmem\_max
  - net.core.wmem\_max
- Linux Auto-tuning buffer limits:
  - net.ipv4.tcp\_rmem
  - net.ipv4.tcp\_wmem
- net.ipv4.tcp\_window\_scaling
- net.ipv4.tcp\_timestamps
- net.ipv4.tcp\_sack



## 7 The Dell Command Utility

The Storage Center SAN can have many of its daily functions managed through a remote command utility called the Dell Command Utility (CompCU). This allows for scripting and automation integration of SAN tasks between the Linux OS and Storage Center. CompCU is a java-packaged application and thus requires the installation of Java on the Linux host. CompCU can be used to script common administrative tasks which can be tremendous time savers and provide a consistent framework for managing Storage Center volumes and Replays.

CompCU requires the host to have the proper Java release installed. Refer to the Command Utility User Guide (in Appendix A) for more details. The CompCU.jar object can be downloaded from the Dell support site. Once installed on the Linux host, this tool can be used to perform Storage Center tasks from the shell prompt, which can be incorporated into new and/or existing user management scripts. Outlined below are some common use cases for CompCU.

- Creating Volumes, mapping to the server.
- Taking replays, recovering replays, etc.

The examples below do not cover the full breadth of the usefulness of CompCU by any means; they are designed to give an initial insight into the types of tasks which may be automated with CompCU.

### 7.1 Verifying Java, configuring and testing CompCU functions

First, install Java (RTE v1.6.x or newer) on the Linux host. The Java runtime may have already been installed with the OS and can be verified with the command shown below.

```
/usr/bin/java -version
java version "1.7.0_07"
Java(TM) SE Runtime Environment (build 1.7.0_07-b10)
Java HotSpot(TM) Server VM (build 23.3-b01, mixed mode)
```

Download the CompCU package from the Dell support site at [http://kc.compellent.com/Published%20Documents/CU060401\\_001.zip](http://kc.compellent.com/Published%20Documents/CU060401_001.zip). The package will include a PDF User Guide as well as the required **CompCU.jar** file. Save this **CompCU.jar** file to a logical file system location. Verify that CompCU is working with Java by executing the command below to display the help and usage syntax.

```
/usr/bin/java -jar ./CompCU.jar -h
Compellent Command Utility (CompCU) 6.4.1.1
```

```
usage: java -jar CompCU.jar [Options] "<Command>"
 -c <arg> Run a single command (option must be within
 quotes)
 -default Saves host, user, and password to encrypted file
```



```

-defaultname <arg> File name to save default host, user, and password
 encrypted file to
-file <arg> Save output to a file
-h Show help message
-host <arg> IP Address/Host Name of Storage Center Management
 IP
-password <arg> Password of user
-s <arg> Run a script of commands
-user <arg> User to log into Storage Center with
-verbose Show debug output
-xmloutputfile <arg> File name to save the CompCU return code in xml
 format. Default is cu_output.xml.

```

[snip]

To facilitate the ease of access in using CompCU, the tool can be initially run with the **-default** switch to configure an encrypted password file as shown below. A file named **default.cli** is created in the local directory. This file may be renamed as required for clarity and usage.

```

/usr/bin/java -jar ./CompCU.jar -default -host 172.16.2.109 -user Admin -
password XXX
Compellent Command Utility (CompCU) 6.4.1.1

```

```

=====
User Name: Admin
Host/IP Address: 172.16.2.109
=====

```

```

Connecting to Storage Center: 172.16.2.109 with user: Admin
java.lang.IllegalStateException: TrustManagerFactoryImpl is not initialized
Saving CompCu Defaults to file [default.cli]...
The "default.cli" file may then be referenced in other commands to login to the
same Storage Center and perform tasks. A separate .cli file may be created for
each Storage Center under management with each containing the appropriate login
credentials for the respective Storage Center array. The example below
demonstrates a "volume show" command applied to the Storage Center located at IP
address 172.16.2.109.

```

```

/usr/bin/java -jar ./CompCU.jar -defaultname default.cli -host 172.16.2.109 -
user Admin -password XXX -c "volume show"
Compellent Command Utility (CompCU) 6.4.1.1

```

```

=====
User Name: Admin
Host/IP Address: 172.16.2.109
Single Command: volume show
=====

```

```

Connecting to Storage Center: 172.16.2.109 with user: Admin
java.lang.IllegalStateException: TrustManagerFactoryImpl is not initialized
Running Command: volume show

```



| Index                                                                          | Name                           | Status          | ConfigSize      | ActiveSize                   |
|--------------------------------------------------------------------------------|--------------------------------|-----------------|-----------------|------------------------------|
| ReplaySize                                                                     | Folder                         |                 |                 |                              |
| StorageProfile                                                                 | DeviceI                        |                 | D               |                              |
| SerialNumber                                                                   | ConfigSizeBlock                | ActiveSizeBlock | ReplaySizeBlock |                              |
| MaxWriteSizeBlo                                                                | ReadCache                      | WriteCache      |                 |                              |
| -----                                                                          |                                |                 |                 |                              |
| -----                                                                          |                                |                 |                 |                              |
| -----                                                                          |                                |                 |                 |                              |
| -----                                                                          |                                |                 |                 |                              |
| -----                                                                          |                                |                 |                 |                              |
| -----                                                                          |                                |                 |                 |                              |
| -----                                                                          |                                |                 |                 |                              |
| 283                                                                            | Fal-asm-mirror-test-failgroup1 | Up              | 100.00 GB       | 31.33 GB                     |
| 0.00 KB                                                                        | Oracle/11gR2/ASM-Mirror        |                 |                 |                              |
| Recommended                                                                    |                                | 6000d31         |                 | 00000650000000000000000012f  |
| 00000065-0000012f                                                              | 209715200                      | 65712128        | 0               | 0                            |
| Enabled                                                                        | Enabled                        |                 |                 |                              |
| 290                                                                            | Fal-asm-mirror-sp-fg1          | Up              | 100.00 GB       | 7.94 GB                      |
| 0.00 KB                                                                        | Oracle/11gR2/ASM-Mirror        |                 |                 |                              |
| Recommended                                                                    |                                | 6000d31         |                 | 000006500000000000000000136  |
| 00000065-00000136                                                              | 209715200                      | 16658432        | 0               | 0                            |
| Enabled                                                                        | Enabled                        |                 |                 |                              |
| 824                                                                            | ibmsvc00-managed-mdisk1        | Up              | 500.00 GB       | 98.02 GB                     |
| 0.00 KB                                                                        | IBMSVC                         |                 |                 |                              |
| Recommended                                                                    |                                | 6000d31         |                 | 000006500000000000000000034c |
| 00000065-0000034c                                                              | 1048576000                     | 205561856       | 0               | 0                            |
| Enabled                                                                        | Enabled                        |                 |                 |                              |
| [snip]                                                                         |                                |                 |                 |                              |
| Successfully finished running Compellent Command Utility (CompCU) application. |                                |                 |                 |                              |

## 7.2 Using CompCU to automate common tasks

This section illustrates some use cases for managing Storage Center tasks with CompCU on Linux. As mentioned above, these examples are indicative of the types of tasks which can easily be accomplished from the Linux shell prompt using CompCU. They are only meant as a starting point to familiarize the system administrator with this powerful tool set.

### 7.2.1 Creating a single volume with CompCU

This example demonstrates using CompCU to create a single 100GB Storage Center volume named **hadrian\_100g\_00** from the Linux host and then placed in the Storage Center **Linux** folder. The volume is mapped to the Linux **hadrian** host.

```
/usr/bin/java -jar ./CompCU.jar -defaultname default.cli -host 172.16.2.109 -
user Admin -password XXX -c "volume create -name hadrian_100g_00 -folder Linux -
server hadrian -size 100g"
```



## 7.2.2 Creating a Replay and a Replay View with CompCU

This example demonstrates with a single CompCU command:

- Creating a Replay, **hadrian\_100g\_00\_Replay** of the existing hadrian\_100g\_00 volume on Storage Center,
- Creating Replay View, **hadrian\_100g\_00\_View** from this mentioned Replay, and
- Mapping the Replay View to the Linux host, **maximus**

```
/usr/bin/java -jar ./CompCU.jar -defaultname default.cli -host 172.16.2.109 -
user Admin -password XXX -c "replay create -volume 'hadrian_100g_00' -name
'hadrian_100g_00_Replay' -view 'hadrian_100g_00_RpView' -server 'maximus'"
```

## 7.2.3 Rapid deployment of multiple volumes with CompCU

This final example demonstrates using CompCU for rapid volume creation and deployment from Storage Center and mapping these volumes to the Linux host, **maximus**.

```
for i in 0 1 2 3 4 5 6 7 8 9; do /usr/bin/java -jar ./CompCU.jar -defaultname
default.cli -host 172.16.2.109 -user Admin -password XXX -c "volume create -name
maximus_10g_0${i} -folder Linux -server 'maximus' -size 10g"; done
```



## A Additional resources

Dell Compellent Storage Center 6.3 Administrator's Guide

<http://kcint.compellent.com/Published%20Documents/680-019-013.pdf>

SC8000 Connectivity Guide

<http://kcint.compellent.com/Knowledge%20Center%20Documents/680-027-013.pdf>

Dell Compellent Storage Center 6.0 Command Utility (CompCU) Reference Guide

<http://kcint.compellent.com/Knowledge%20Center%20Documents/680-018-007.pdf>

Red Hat Enterprise Linux Document Portal

[https://access.redhat.com/site/documentation/Red\\_Hat\\_Enterprise\\_Linux/?locale=en-US](https://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/?locale=en-US)



## B Configuration details

Table 1 Component table

| Component        | Description                                                     |
|------------------|-----------------------------------------------------------------|
| Operating system | Red Hat Enterprise Linux Server release 6.4 (Santiago)          |
| Driver version   | Driver Version = 8.04.00.04.06.3-k-debug<br>BIOS Version = 3.00 |
| Firmware version | Firmware Version = 5.06.05 (90d5)                               |
| Application      | NA                                                              |
| Cabling          | 4GB Fibre Channel                                               |
| Server           | Dell R810                                                       |
| Storage          | Dell Storage Center SCOS 6.4.10, Virtual port mode              |
| Switch           | Cisco MDS 9148                                                  |

